AFRL-VA-WP-TP-2004-308

UAV TASK ASSIGNMENT WITH TIMING CONSTRAINTS VIA MIXED-INTEGER LINEAR PROGRAMMING

Corey Schumacher, Phillip Chandler, Meir Pachter, and Lior Pachter

FEBRUARY 2004

Approved for public release; distribution is unlimited.

20040324 066

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From - To)
February 2004	Conference Paper Preprint	

4. TITLE AND SUBTITLE	
UAV TASK ASSIGNMENT WITH TIMING CONSTRAINTS VIA MIXED-INTEGER LINEAR PROGRAMMING	5a. CONTRACT NUMBER — In-house
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER — N/A

6. AUTHOR(S)	
Corey Schumacher and Phillip Chandler (AFRL/VACA)	5d. PROJECT NUMBER — N/A
Meir Pachter (AFIT/ENG)	5e. TASK NUMBER — N/A
Lior Pachter (University of California at Berkeley)	5f. WORK UNIT NUMBER — N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER
Control Theory Optimization Branch (AFRL/VACA) Control Sciences Division Air Vehicles Directorate Air Force Research Laboratory, Air Force Materiel Command Wright-Patterson AFB, OH 45433-7542	AFIT/ENG University of California at Berkeley	AFRL-VA-WP-TP-2004-308

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY ACRONYM(S)
Air Vehicles Directorate Air Force Research Laboratory Air Force Materiel Command Wright-Patterson Air Force Base, OH 45433-7542	AFRL/VACA
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) — AFRL-VA-WP-TP-2004-308

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The optimal timing of air-to-ground tasks is undertaken. Specifically, a scenario where multiple air vehicles are required to prosecute geographically dispersed targets is considered. The vehicles must perform multiple tasks on each target. The targets must be found, classified, attacked, and verified as destroyed. The optimal performance of these tasks requires cooperation amongst the vehicles such that critical timing constraints are satisfied. In this paper, an optimal task assignment and timing algorithm is developed, using a mixed integer linear program, or MILP, formulation. MILP can be used to assign all tasks to the vehicles in an optimal manner, including variable arrival times, for groups of air vehicles with coupled tasks involving timing and task order constraints. When the air vehicles have sufficient endurance, the existence of a solution is guaranteed.

15. SUBJECT TERMS
Optimization, timing, coupled tasks, autonomous control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT — Unclassified	b. ABSTRACT — Unclassified	c. THIS PAGE — Unclassified	SAR	20	Corey Schumacher — 19b. TELEPHONE NUMBER (Include Area Code) (937) 255-8494

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

UAV Task Assignment with Timing Constraints via Mixed-Integer Linear Programming

Corey Schumacher
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, OH 45433-7531

Phillip Chandler
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, OH 45433-7531

Meir Pachter
Department of Electrical Engineering
Air Force Institute of Technology (AFIT/ENG)
Wright-Patterson AFB, OH 45433-7765

Lior Pachter
Department of Mathematics
University of California at Berkeley
Berkeley, CA 94720-3840

Abstract

The optimal timing of air-to-ground tasks is undertaken. Specifically, a scenario where multiple air vehicles are required to prosecute geographically dispersed targets is considered. The vehicles must perform multiple tasks on each target. The targets must be found, classified, attacked, and verified as destroyed. The optimal performance of these tasks requires cooperation amongst the vehicles such that critical timing constraints are satisfied. In this paper, an optimal task assignment and timing algorithm is developed, using a mixed integer linear program, or MILP, formulation. MILP can be used to assign all tasks to the vehicles in an optimal manner, including variable arrival times, for groups of air vehicles with coupled tasks involving timing and task order constraints. When the air vehicles have sufficient endurance, the existence of a solution is guaranteed.

1 Introduction

Autonomous wide area search munitions (WASM) are small, powered unmanned air vehicles (UAV's), each with a turbojet engine and sufficient fuel to fly for a short period of time. They are deployed in groups, or "swarms," from larger aircraft flying at higher altitudes. They are typically deployed in groups of four, although larger swarms are certainly possible. They are individually capable of searching for, recognizing, and attacking targets. Cooperation between munitions has the potential to greatly improve their effectiveness in many situations. The ability to communicate target information to one another will greatly improve the capability of future search munitions.

In [1-3], a time-phased network optimization model was used to perform task allocation for a group of powered munitions. The model is run simultaneously on all munitions at discrete points in time, and assigns each vehicle one or more tasks each time it is run. The model is solved each time new information is brought into the system, typically because a new target has been discovered or an already-known target's status has been changed. The network optimization model is run iteratively so that all of the known targets will be completely prosecuted by the resulting allocation. Classification, attack, and battle damage assessment tasks can all be assigned to different vehicles when a target is found, resulting in the target being more quickly serviced. A single vehicle can also be given multiple task assignments to be performed in succession, if that is more efficient than having multiple vehicles perform the tasks individually. In [2], variable path lengths are added to guarantee that feasible trajectories will be calculated for all tasks. This method is computationally efficient and can quickly assign all of the needed tasks to the available vehicles, even for large numbers of vehicles, however the iterative procedure is heuristic and does not guarantee that the solution is near optimal.

This paper proposes an optimal formulation for solving the coupled multiple-assignment problem. Formulating the problem in a Mixed Integer Linear Program (MILP) format will allow the optimal solution to be found while satisfying all timing constraints. The inclusion of continuous timing variables allows solutions with any feasible task completion times to be calculated. The MILP formulation for task assignment proposed was first presented in [4]. The initial formulation presented in [4] did not include some of the extensions and potential applications discussed here. Most importantly, [4] proposed the general methodology and architecture, but did not include solutions for meaningfully-sized problems, which this paper presents. In this paper, time is treated as a continuous variable and a rigorous optimal task assignment algorithm is developed. This requires the solution of a mixed integer linear program [6].

2 Scenario

Assume there is a number of wide area search munitions or air vehicles searching an area for unknown targets. Typically, four munitions will be deployed as a team, although multiple pods of four could be linked together to form a larger team. Vehicles travel in a pre-specified zamboni search pattern with a sensor that is capable of detecting and identifying potential targets. As the vehicles search, they come across enemy targets. When a potential target is discovered, it is necessary to have a second vehicle examine the target and classify it, to be confident that it is a valid target, and the target must then be attacked and verified as having been destroyed. The most common occurrence in this scenario is to find an individual target, and require the assignment of up to three vehicles to fully prosecute the target. That situation will be demonstrated in this paper. Having a larger number of air vehicles involved, or finding multiple targets in quick succession, would result in a larger assignment problem. This situation is addressed in a suboptimal manner in [1-3]. Here, we present an optimal mixed-integer linear programming solution for the n-target w-vehicle assignment problem, as well as a solution example for the 1-target 3-vehicle case.

Suppose we have n geographically dispersed targets with unknown position and w air vehicles (AV). We assume $w \geq n + 1$. We then have $n+w+1$ nodes: n target nodes, w source (or vehicle) nodes, and one sink node. Nodes $1,...,n$ are located at the n target positions. Nodes $n+1,...n+w$ are located at the vehicle initial positions. Nodes $n+w+1$ is the "sink". An air vehicle with no future target assignments is relegated to the sink, and will continue to search. A vehicle located at the sink cannot be reassigned during the present assignment computation.

Spatial outlay: The flight time of AV v from node i to node j is $t_{ij}^{(v,k)} \geq 0$. The indices $i=1,...,n+w$, $j=1,...n$, and $v = 1,...,w$. The index k designates the task to be performed at node j. The time to travel from node i to node j depends on the particular AV's airspeed and the assigned task k.

The tasks: Three tasks must be performed on each target.
1. $k=1$ – Classification

2. $k=2$ – Attack

3. $k=3$ – Target Damage Assessment (Verification)

Furthermore, once an AV attacks a target, it is destroyed and can no longer perform additional tasks. This is certainly the case for powered munitions, but if the AV is a reusable aircraft, one has to account for the depletion of its store of ammunition following each attack.

The three tasks must be performed on each target in the order listed. This results in critical timing constraints, which set this problem apart from the classical Vehicle Routing Problem (VRP). In the latter, rigid time windows for the arrival of the vehicles can be specified, however, the coupling brought about by the need to schedule the various tasks is absent. Evidently, our problem features some aspects of job shop scheduling.

In the operational scenario considered, the number of problem parameters $t_{ij}^{(v,k)}$ is $3wn+3n(n-1)w = 3n^2w$. When Euclidean distances are used, the dimension of the parameter space is reduced to $0.5n(n-1)+wn= 0.5n(n+2w-1)$. Finally, the endurance of AV v is $T_v, v=1,...,w$.

Figure 1 illustrates a scenario where 1 stationary ground target is engaged by three AVs. The potential target's position is known at the beginning of the optimization, but not the classification.

3 MILP Model

The MILP model uses a discrete approximation of the real world based on nodes that represent discrete start and end positions for segments of a UAVs path. Nodes representing target positions range from $1...n$ and nodes for UAV positions range from $1+n...w+n$. There is also an additional logical node for the sink $n+w+1$. The sink node is used when a UAV is not assigned to attack a target; it goes to the sink when it is done with all of its tasks, or when it is not assigned another task. When a UAV enters the sink it is then used for searching the battlefield. The MILP model requires the minimum costs or times for a UAV to fly from one node to another node. We assume that any flight time larger than these minimum times is continuously achievable. These flight times are constants represented by

$t_{ij}^{(v,k)}$, the time it takes UAV v to fly from node i to node j to perform task k. The flight times are positive real numbers, $t_{ij}^{(v,k)} \geq 0$.

3.1 Decision Variables

The decision variables $x_{ij}^{(v,k)} = 1$ if AV v is assigned to fly from node i to node j and perform task k at node j, and 0 otherwise; $i = 1,...n+w$, $j = 1,...,n$, $v = 1,...,w$, and $k = 1,2,3$. For task assignments $k=1,3$, $i \neq j$ and for task assignment $k=2$ we allow $i = j$; the latter allows for an AV to perform the target classification task, and immediately thereafter attack the target. Thus far, we have $wn(3n+1)$ binary decision variables.

We also have the following additional binary decision variables. The decision variable $x_{i,n+w+1}^{(v)} = 1$ if AV v is assigned to fly from node i to the sink $n+w+1$, and is 0 otherwise; $v=1,...,w$ and $i = 1,...,n+w$. This adds $(n+1)w$ binary decision variables. Entering the sink can also be thought of as being assigned to the search task.

Continuous decision variables:

The time of performance of task k on target j is $t_j^{(k)} > 0$; $k = 1,2,3$ and $j = 1,...,n$. Thus, we have $3n$ continuous decision variables. We also have w additional continous decision variables: the time AV v leaves node $j = n + v$ is t_v; $v = 1,...,w$. In total we have $w[n(3n+2)+1]$ binary decision variables and $3n+w$ continuous non-negative decision variables.

3.2 Cost Functions

A variety of cost functions are possible, depending on the exact application, and other variations in the problem formulation. Possible cost functions include:

1. Minimize the total flight time of the AVs

$$J = \sum_{k=1}^{3} \sum_{v=1}^{w} \sum_{i=1}^{n+w} \sum_{j=1}^{n} t_{i,j}^{(v,k)} x_{i,j}^{(v,k)} \tag{1}$$

2. Alternatively, minimize the total engagement time. The target j is visited for the last time at time $t_j^{(3)}$. Let t_f be the time at which all targets have been through Verification. Introduce an additional continuous decision variable $t_f \in \Re_+^1$. The cost function is then $J = t$ and we minimize J subject to the constraints

$$t_j^{(3)} \leq t_f, j = 1,...,n \tag{2}$$

We also add a small weight to the time of performance of each individual task, to encourage each individual task to be completed as quickly as possible. Then

$$J = t_f + c_j^{(k)} t_j^{(k)}, j = 1,...,n, k = 1,2,3, \tag{3}$$

where $c_j^{(k)} > 0$ is a small weight on the completion time of each individual task. To weight the time of performance of individual tasks more heavily, one could use $J = c_j^{(k)} t_j^{(k)}, j = 1,...,n, k = 1,2,3$.

3. Other cost functions could also be formulated. For example, the problem could be formulated to maximize a benefit function, similar to that used in [1-3]. This would allow direct incorporation of competing search tasks, if all target tasks were not required to be completed.

3.3 Constraints

The formulation of the MILP is primarily based on the constraints. Proper enumeration of all of the required constraints is critical to achieving the desired vehicle behavior.

1. Mission completion requires that all three tasks are performed on each target exactly one time:

$$\sum_{v=1}^{w}\sum_{i=1,i\neq j}^{n+w} x_{ij}^{(v,k)}=1, \quad \begin{matrix} k=1,3 \\ j=1,...,n \end{matrix}, \tag{4}$$

and

$$\sum_{v=1}^{w}\sum_{i=1}^{n+w} x_{ij}^{(v,2)}=1, \quad j=1,...,n \tag{5}$$

This yields *3n* constraints.

2. Not more than one AV is assigned to perform a specific task *k* on a specified target *j*:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,k)}\leq 1, \quad v=1,...,w; \tag{6}$$

where *j=1,...,n* and *k=1,3*, and

$$\sum_{i=1}^{n+w} x_{i,j}^{(v,2)}\leq 1, \quad v=1,...,w; j=1,...,n \tag{7}$$

This yields *3n* constraints. This constraint is redundant with Constraint 1, and will not be included in the examples. However, this constraint could be important with modifications to the cost function and Constraint 1.

3. An AV *v*, coming from the outside, can visit target *j* at most once:

$$\sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n+w} x_{ij}^{(v,k)}\leq 1, \quad \begin{matrix} v=1,...,w \\ j=1,...,n \end{matrix} \tag{8}$$

This, and Condition 4 below, are simplifying assumptions that eliminate the possibility of loops. In addition, each AV *v* can only enter the sink once:

$$\sum_{i=1}^{n+w} x_{i,n+w+1}^{(v)}\leq 1, \quad v=1,...,w \tag{9}$$

This yields *(n+1)w* constraints.

4. AV *v* leaves node *j* at most once:

$$\sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{ij}^{(v,k)}+x_{j,n+w+1}^{(v)}\leq 1, \quad v=1,...,w; \tag{10}$$

where *j=1,...,n*. This yields *nw* constraints.

5 A munition is perishable. An AV *v* can be assigned to attack at most one target. Thus,

$$\sum_{j=1}^{n}\sum_{i=1}^{n+w} x_{ij}^{(v,2)}\leq 1, \quad \forall v=1,...,w. \tag{11}$$

This yields *w* constraint equations.

1. If AV *v* is assigned to fly to target *j* for Verification, it cannot possibly be assigned to attack target *j*:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,2)}\leq 1-\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,3)}, \quad v=1,...,w; \tag{12}$$

where *j = 1,...,n*. Condition 3 renders Condition 6 redundant; we do however include Condition 6, because it holds in its own right, but also in the case where one would choose not to have recourse to the Simplifying Assumption which yields Condition 3.

2. Continuity 1. If AV v enters target (node) j for the purpose of performing task 3, it must also exit target j:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,3)}\leq \sum_{k=1}^{3}\sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)}+x_{j,n+w+1}^{(v)}, \tag{13}$$

where *j=1,...,n; v=1,...,w.*

Continuity 2. If AV v enters target (node) j for the purpose of performing task 1, it must also "exit" target (node) j:

$$\sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,1)} \leq \sum_{k=1}^{3} \sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j,j}^{(v,2)} + x_{j,n+w+1}^{(v)}, \qquad \text{where } j=1,...,n; \ v=1,...,w. \tag{14}$$

Continuity 3. A munition is perishable. Thus, if AV v is assigned to fly to target (node) j to perform task $k=2$, then, at any other point in time, AV v cannot also be assigned to fly from target j to a target i, $i \neq j$, to perform any other tastk at target i; recall that according to our Simplifying Assumption, AV v can enter target j not more than once. Thus

$$\sum_{k=1}^{3} \sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j,n+w+1}^{(v)} \leq 1 - \sum_{i=1}^{n+w} x_{i,j}^{(v,2)}, \tag{15}$$

where $j=1,...,n; \ v=1,...,w.$

Continuity 4. If AV v is not assigned to visit node j, then it cannot possibly be assigned to fly out of node j. Thus

$$\sum_{k=1}^{3} \sum_{i=1,i\neq j}^{n} x_{j,i}^{(v,k)} + x_{j,n+w+1}^{(v)} \leq \sum_{k=1}^{3} \sum_{i=1,i\neq j}^{n+w} x_{i,j}^{(v,k)}, \qquad \text{where } j=1,...,n; \ v=1,...,w. \tag{16}$$

Continuity 5. All AVs leave the source nodes. An AV leaves the source node even if this entails a direct assignment to the sink.

$$\sum_{k=1}^{3} \sum_{j=1}^{n} x_{n+v,j}^{(v,k)} + x_{n+v,n+w+1}^{(v)} = 1, \quad \forall v=1,...,w. \tag{17}$$

Continuity 6. An AV cannot attack target (node) i, coming from target (node) i, unless it entered target (node) i to perform a classification. Thus

$$x_{i,i}^{(v,2)} \leq \sum_{j=1}^{n+w} x_{j,i}^{(v,1)}, \forall \begin{array}{l} i=1,...,n \\ v=1,...,w \end{array}. \tag{18}$$

3. Timing Constraints

Nonlinear equations which enforce the timing constraints are easily derived, and are given in [4]. We are however interested in an alternative formulation which uses linear inequalities. Thus, let

$$T \equiv \max_v \{T_v\}_{v=1}^{w}. \tag{19}$$

Then the linear timing constraints become:

$$t_j^{(k)} \leq t_i^{(1)} + t_{i,j}^{(v,k)} + \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,1)}\right) wT \tag{20}$$

$$t_j^{(k)} \geq t_i^{(1)} + t_{i,j}^{(v,k)} - \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,1)}\right) wT \tag{21}$$

$$t_j^{(k)} \leq t_i^{(3)} + t_{i,j}^{(v,k)} + \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,3)}\right) wT \tag{22}$$

$$t_j^{(k)} \geq t_i^{(3)} + t_{i,j}^{(v,k)} - \left(2 - x_{i,j}^{(v,k)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,3)}\right) wT \tag{23}$$

for $i=1,...,n; \ j=1,...,n; \ i \neq j; v=1,...w; \ k=1,3.$ In addition,

$$t_j^{(2)} \leq t_i^{(1)} + t_{i,j}^{(v,2)} + \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1,l\neq i}^{n+w} x_{l,i}^{(v,1)}\right) wT \tag{24}$$

$$t_j^{(2)} \geq t_i^1 + t_{i,j}^{(v,2)} - \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1, l \neq i}^{n+w} x_{l,i}^{(v,1)}\right) wT \qquad (25)$$

$$t_j^{(2)} \leq t_i^{(3)} + t_{i,j}^{(v,2)} + \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1, l \neq i}^{n+w} x_{l,i}^{(v,3)}\right) wT \qquad (26)$$

$$t_j^{(2)} \geq t_i^{(3)} + t_{i,j}^{(v,2)} - \left(2 - x_{i,j}^{(v,2)} - \sum_{l=1, l \neq i}^{n+w} x_{l,i}^{(v,3)}\right) wT \qquad (27)$$

for $i=1,...,n; j=1,...,n; i \neq j ; v=1,...w.$

Also,

$$t_j^{(k)} \leq t_v + t_{n+v,j}^{(v,k)} + \left(1 - x_{n+v,j}^{(v,k)}\right) wT \qquad (28)$$

$$t_j^{(k)} \geq t_v + t_{n+v,j}^{(v,k)} - \left(1 - x_{n+v,j}^{(v,k)}\right) wT \qquad (29)$$

for all $j=1,...,n; k = 1,2,3; v = 1,...,w.$

These timing constraints operate in pairs. They are loose inequalities which do not come into play for assignments $x_{i,j}^{(v,k)}$ which do not occur, but effectively become hard equality constraints for assignments which do occur. Thus the time that a task k is performed on target j by AV v will be equal to the time that the preceding task was performed by AV v at node i, plus the time it will take AV v to fly from node i to node j. A similar constraint applies if AV v left its source node $n+v$ to fly to node j.

Furthermore,

$$t_j^{(1)} \leq t_j^2, \forall j = 1,...,n \qquad (30)$$

$$t_j^{(2)} < t_j^3, \forall j = 1,...,n \qquad (31)$$

The timing constraints thus add $2n[(6n-1)w+1]$ linear inequality constraints.

3.4 Extensions

Additional constraints can be included.

9. A vehicle's assigned path cannot be longer than its remaining endurance T_v:

$$\sum_{k=1}^{3} \sum_{i=1}^{n+w} \sum_{j=1, j \neq i}^{n} t_{i,j}^{(v,k)} x_{i,j}^{(v,k)} \leq T_v, \qquad v = 1,...,w \qquad (32)$$

This yields w constraints.

10. It is fairly easy to specify additional rigid time window constraints akin to the VRP, e.g., for time critical targets one could demand that the attack on target j take place after time $\bar{t}_j^{(2)}$, and not before time $\underline{t}_j^{(2)}$, i.e.

$$\underline{t}_j^{(2)} \leq t_j^{(2)} \leq \bar{t}_j^{(2)}, \qquad \forall j = 1,...,n \qquad (33)$$

11. Numerous other constraints can also be included, such as: specific vehicles performing certain tasks, minimum time delays between tasks, simultaneous completion of attack tasks, and requiring the vehicle that classifies a target to also attack it. With some constraints, such as vehicle endurance (Constraint 9), the existence of a solution is no longer guaranteed.

12. Hetergeneous vehicles: For some applications, a set of heterogeneous vehicles would be used, with different capabilities. Some might be sensor platforms with no attack capability. Or some vehicles might simply have used all their ordinance, or not be carrying the proper ordinance to attack certain targets. In such cases, we add the constraint $x_{i,j}^{(v,k)} = 0$ for any combination where vehicle v cannot perform task k on target j.

13. Partially prosecuted targets: If this algorithm was used for task assignment by a group of UAV's, additional targets and tasks could be added to the overall task list while some previously-known targets were already partly prosecuted. In this case, less than three tasks would be required for some targets, when the assignments were recalculated. For already completed tasks, we modify Constraint 1 such that

$$\sum_{v=1}^{w}\sum_{i=1,i\neq j}^{n+w} x_{ij}^{(v,k)} = 0, \quad \begin{array}{l} k=1,3 \\ j=1,...,n \end{array}, \tag{34}$$

and

$$\sum_{v=1}^{w}\sum_{i=1}^{n+w} x_{ij}^{(v,2)} = 0, \quad j=1,...,n \tag{35}$$

for any target j and task k that have already been completed.

4 Problem Solution

4.1 One Target and Three Avs

We will first consider the case of one target and three AVs, i.e. $n=1$ and $w=3$, as the problem is small enough to be described in detail. In this case, we have 18 binary decision variables and 6 continuous decision variables. Minimizing the time the final task occurs will add an additional continuous decision variable t, for a total of 25 decision variables. In this single-target case, we could exclude the additional variable and simply minimize t_1^3, but the additional variable will be included to demonstrate the additional variable which would be required for $n \geq 2$. An example State Transition Diagram is given in Figure 1, for the $n=2$, $w=3$ case.

There are 18 binary decision variables:
$$\begin{aligned} (x_1,...,x_5) &= \left(x_{1,1}^{(1,2)}, x_{1,1}^{(2,2)}, x_{1,1}^{(3,2)}, x_{2,1}^{(1,1)}, x_{2,1}^{(1,2)} \right), \\ (x_6,...,x_{10}) &= \left(x_{2,1}^{(1,3)}, x_{3,1}^{(2,1)}, x_{3,1}^{(2,2)}, x_{3,1}^{(2,3)}, x_{4,1}^{(3,1)}, \right) \end{aligned} \tag{36}$$

$$(x_{11},...,x_{15}) = \left(x_{4,1}^{(3,2)}, x_{4,1}^{(3,3)}, x_{1,5}^{(1)}, x_{1,5}^{(2)}, x_{1,5}^{(3)} \right)$$

$$(x_{16},x_{17},x_{18}) = \left(x_{2,5}^{(1)}, x_{3,5}^{(2)}, x_{4,6}^{(3)} \right)$$

There are 7 continuous decision variables:
$$(x_{19},...,x_{25}) = \left(t_1, t_2, t_3, t_1^{(1)}, t_1^{(2)}, t_1^{(3)}, t \right) \tag{37}$$

We wish to minimize
$$J = x_{25} + 0.1(x_{22} + x_{23} + x_{24}) \tag{38}$$
subject to the following constraints:

From Constraint 1:
$$x_4 + x_7 + x_{10} = 1$$
$$x_6 + x_9 + x_{12} = 1 \tag{39}$$
$$x_5 + x_8 + x_{11} + x_1 + x_2 + x_3 = 1$$

From Constraint 7.5:
$$x_4 + x_5 + x_6 + x_{16} = 1$$
$$x_7 + x_8 + x_9 + x_{17} = 1 \tag{40}$$
$$x_{10} + x_{11} + x_{12} + x_{18} = 1$$

Thus we have 6 equality constraints, plus the following inequality constraints.
From Constraint 2:

$$x_4 + x_7 + x_{10} \le 1$$
$$x_6 + x_9 + x_{12} \le 1 \tag{41}$$
$$x_5 + x_8 + x_{11} + x_1 + x_2 + x_3 \le 1$$

From Constraint 3:
$$x_4 + x_5 + x_6 \le 1$$
$$x_7 + x_8 + x_9 \le 1 \tag{42}$$
$$x_{10} + x_{11} + x_{12} \le 1$$

Constraints 4-6 drop out in the 1-target case.
The Continuity Constraints give:
From 7.1:

$$x_6 \le x_{13}$$

$$x_9 \le x_{14} \tag{43}$$

$$x_{12} \le x_{15}$$

From 7.2:
$$x_4 \le x_1 + x_{13}$$

$$x_7 \le x_2 + x_{14} \tag{44}$$

$$x_{10} \le x_3 + x_{15}$$

From 7.3:
$$x_{13} + x_1 + x_5 \le 1$$
$$x_{14} + x_2 + x_8 \le 1 \tag{45}$$
$$x_{15} + x_3 + x_{11} \le 1$$

From 7.4:
$$x_{13} \le x_4 + x_6$$

$$x_{14} \le x_7 + x_9 \tag{46}$$

$$x_{15} \le x_{10} + x_{12}$$

From 7.5:
The equality constraints given by Eq. 36.

From 7.6:
$$x_1 \le x_4$$

$$x_2 \le x_7 \tag{47}$$

$$x_3 \le x_{10}$$

With only 1 target node, the Constraints associated with Eq (20-23) and (26,27) are not meaningful. So we are left with the following timing constraints:

From Eq (24,25):
$$x_{23} \le x_{22} + t_{1,1}^{(1,2)} + (2 - x_1 - x_4)wT$$
$$x_{23} \ge x_{22} + t_{1,1}^{(1,2)} - (2 - x_1 - x_4)wT$$

$$x_{23} \leq x_{22} + t_{1,1}^{(2,2)} + (2 - x_2 - x_7)wT$$

$$x_{23} \geq x_{22} + t_{1,1}^{(2,2)} - (2 - x_2 - x_7)wT$$

$$x_{23} \leq x_{22} + t_{1,1}^{(3,2)} + (2 - x_3 - x_{10})wT$$

$$x_{23} \geq x_{22} + t_{1,1}^{(3,2)} - (2 - x_3 - x_{10})wT$$

(48)

From Eq (28,29):

$$x_{22} \leq x_{19} + t_{2,1}^{(1,1)} + (1 - x_4)wT$$

$$x_{22} \geq x_{19} + t_{2,1}^{(1,1)} - (1 - x_4)wT$$

$$x_{22} \leq x_{20} + t_{3,1}^{(2,1)} + (1 - x_7)wT$$

$$x_{22} \geq x_{20} + t_{3,1}^{(2,1)} - (1 - x_7)wT$$

$$x_{22} \leq x_{21} + t_{4,1}^{(3,1)} + (1 - x_{10})wT$$

$$x_{22} \geq x_{21} + t_{4,1}^{(3,1)} - (1 - x_{10})wT$$

(49)

and

$$x_{23} \leq x_{19} + t_{2,1}^{(1,2)} + (1 - x_5)wT$$

$$x_{23} \geq x_{19} + t_{2,1}^{(1,2)} - (1 - x_5)wT$$

$$x_{23} \leq x_{20} + t_{3,1}^{(2,2)} + (1 - x_8)wT$$

$$x_{23} \geq x_{20} + t_{3,1}^{(2,2)} - (1 - x_8)wT$$

$$x_{23} \leq x_{21} + t_{4,1}^{(3,2)} + (1 - x_{11})wT$$

$$x_{22} \geq x_{21} + t_{4,1}^{(3,2)} - (1 - x_{11})wT$$

(50)

and finally,

$$x_{24} \leq x_{19} + t_{2,1}^{(1,3)} + (1 - x_6)wT$$

$$x_{24} \geq x_{19} + t_{2,1}^{(1,3)} - (1 - x_6)wT$$

$$x_{24} \leq x_{20} + t_{3,1}^{(2,3)} + (1 - x_9)wT$$

$$x_{24} \geq x_{20} + t_{3,1}^{(2,3)} - (1 - x_9)wT$$

$$x_{24} \leq x_{21} + t_{4,1}^{(3,3)} + (1 - x_{12})wT$$

$$x_{24} \geq x_{21} + t_{4,1}^{(3,3)} - (1 - x_{12})wT$$

(51)

Also, from Eq (30,31), we have:

$$x_{22} \leq x_{23} - \varepsilon$$

(52)

9

and

$$x_{23} \leq x_{24} - \varepsilon \tag{53}$$

where $\varepsilon > 0$ is a small positive constant. We will set $\varepsilon = 0.1$. This enforces a small delay between each task being performed on a target.

Finally, from Eq (2), we have

$$x_{24} \leq x_{25}. \tag{54}$$

Thus the full set of constraints contains 6 equality constraints and 51 inequality constraints, for 57 total constraints. A few of them are redundant for this case, but might not be for a more complex problem.

Let us make the simplifying assumption that the time to travel from node i to node j to perform task k is independent of which task is required, and which vehicle is performing the task. Then $t_{i,j}^{(v,k)}$ simply becomes $t_{i,j}$. For this example, let

$t_{1,1} = 0.1$

$t_{2,1} = 3.61$

$t_{3,1} = 4.24$

$t_{4,1} = 5.39$

We will arbitrarily set $T = Tv = 100$ as the endurance of all of the AVs, so that endurance is not a constraint. Then the optimal assignment is:
$x_i = 1, i=1,4, 9,14,18$
$x_i = 0, i=2,3,5,...,8,10,...,13,15,16,17$
$x_i = 0, i=19,20,21.$
$x_{22} = 3.61$
$x_{23} = 3.71$
$x_{24} = 4.24$
$x_{25} = 4.24$

This corresponds with all 3 vehicles immediately leaving their source nodes (x_{19}-x_{21}=0), and vehicle 1 performing classify and attack on the target at $t=3.61$ and 3.71 respectively, with vehicle 2 performing verification at $T=4.24$. Vehicle 3 flies direction to the sink (it is not assigned to this target, but continues to search).

Suppose that it takes longer for a vehicle that has just classified a target to complete an attack on that target. Then we might have the initial conditions
$t_{1,1} = 1$

$t_{2,1} = 3.61$

$t_{3,1} = 4.24$

$t_{4,1} = 5.39$

In this case, the assignment is identical, except that the attack occurs at $t=4.61$ and the verification at $t=4.71$. This is an example where the verification had to be delayed so that it occurred after the attack.

Finally, suppose that Vehicle 3 is closer to the target initially, and we have the initial conditions
$t_{1,1} = 1$

$t_{2,1} = 3.61$

$t_{3,1} = 4.24$

$t_{4,1} = 4.50$

Then the optimal assignment is:
$x_i = 1, i=4,8,12,13,15$

10

$x_i = 0$, $i=1,2,3,56,7,9,10,11,14,16,17,18$
$x_i = 0$, $i=19,20,21$.
$x_{22} = 3.61$
$x_{23} = 4.24$
$x_{24} = 4.50$
$x_{25} = 4.50$

This assignment requires all 3 vehicles to immediately leave their source nodes and proceed to the target on minimum-time trajectories. Vehicle 1 performed the classification, vehicle 2 the attack, and vehicle 3 the verification. Vehicles 1 and 3 then proceed to the sink (and continue to search for other targets).

4.2 Larger Size Problems

The size of the optimization problem expands rapidly as problem size increases. However, some practically-sized problems are amendable to optimal solution with this mixed-integer linear program formulation. For n targets, w vehicles, and $m=3$ tasks per target, the problem size scales as follows: There are $n(n-1)wm+nwm+2nw+mn+2w+1$ decision variables. Of these, $3+nm+1$ are continuous timing variables, and the rest are binary decision variables. The number of constraints likewise grows exponentially. There are $12*(n-1)*n*w+9*n*w+2*n*w*m+2*n*m+3*w$ constraints. Of these, $m*n+w$ are equality constraints. The rest are inequality constraints, including $7*n*w+2*w$ inequality non-timing constraints, and $12*(n-1)*n*w+2*n*w+2*n*w*m+m*n$ inequality timing constraints.

For n=2, w=3, there are 51 binary decision variables, 10 continuous decision variables, 9 linear equality constraints, and 174 linear inequality constraints. For n=2, w=4, there are 68 binary decision variables, 11 continuous decision variables, 10 linear equality constraints, and 230 linear inequality constraints. For n=2, w=5, there are 85 binary decision variables, 12 continuous decision variables, 11 linear equality constraints, and 286 linear inequality constraints. Problem size and complexity grow much more rapidly with an increased number of targets. For n=3, w=4, there are 136 binary decision variables, 14 continuous decision variables, 13 linear equality constraints, and 485 linear inequality constraints. The growth of constraints and variables is linear in the number of vehicles, but quadratic in the number of targets.

The initial condition of a problem is defined by the relative distances, or flight times, between the nodes. For the following examples, let T_{ij} be defined as:

$$T_{ij} = \begin{matrix} 0.2000 & 5.8310 & 7.0711 \\ 5.8310 & 0.2000 & 7.2111 \\ 7.0711 & 7.2111 & 0.2000 \\ 7.2801 & 5.0000 & 3.0000 \\ 9.2195 & 3.6056 & 8.0623 \\ 3.1623 & 8.4853 & 10.000 \\ 10.000 & 8.6023 & 3.1623 \end{matrix}$$

where the start node i is indexed down the rows, and the end node j is indexed over the columns. So the time for a vehicle to fly from node 4 to node 3 $T_{43} = 3.0$, and so on. The diagonal elements, $i=j$, correspond with a vehicle performing an attack on a target immediately after classifying it, and thus include only a small delay. The sink position does not matter, as it only exists conceptually. The time to reach it would not be meaningful. A vehicle that is assigned to the sink continues to search for potential targets along a predefined search path.

For $n=2$, $v=3$, the optimal solution is for: Vehicle 1 to classify Target 1 at $t=7.1$ and attack Target 1 at $t=7.3$, Vehicle 3 to classify Target 2 at $t=3.6$, and attack Target 2 at $t=3.8$, and for Vehicle 2 to verify Target 2 at $t=5.0$, and verify Target 1 at $t=10.8$. This solution has an optimal minimum cost of $J=14.6$. All tasks are completed by t=10.8. This optimization problem took 1.06 seconds to solve using GnuTools on a 700MHz Pentium II processor.

For $n=2$, $v=4$, the optimal solution is for: Vehicle 4 to classify Target 1 at $t=3.2$ and attack Target 1 at $t=3.4$, Vehicle 3 to classify Target 2 at $t=3.6$, and attack Target 2 at $t=3.8$, Vehicle 1 to verify Target 1 at $t=7.1$, and Vehicle 2 to verify Target 2 at $t=5.0$. This solution has an optimal minimum cost of $J=9.67$. All tasks are completed by $t=7.1$. In this case, the availability of an additional vehicle which started out closer to Target 1 allowed the tasks

to be completed much more quickly. This optimization problem took 5.17 seconds to solve using GnuTools on a 700MHz Pentium II processor. The $n=2$, $v=5$ solution is identical, with the addition that vehicle 5 is assigned directly to the sink. All of the target tasks are completed by the same Vehicles, at the same times, as in the $n=2$, $v=4$ case. The n=2, v=5 case is solved using GnuTools on a 700MHz Pentium II processor in 8.4 seconds.

For $n=3$, $v=4$, the problem is starting to become quite complex. The optimal solution is for: Vehicle 2 to classify Target 2 at $t=3.6$ and attack Target 2 at $t=3.8$, Vehicle 1 to classify Target 3 at $t=3.0$, and attack Target 3 at $t=3.2$, Vehicle 4 to delay 0.14, then verify Target 3 at $t=3.3$, and then to classify Target 1 at $t=10.4$ and attack Target 1 and $t=10.6$, and for Vehicle 3 to verify Target 2 at $t=8.5$, and then verify Target 1 at $t=14.3$. This solution has an optimal minimum cost of $J=20.4$. All tasks are completed by $t=14.3$. In this case, Vehicle 4 exhibits a non-zero path length extension (delay) before beginning its set of tasks. If it had immediately proceeded to its first task without delay, it would have performed verification on Target 3 before the attack had occurred. For this case, the optimization problem took 27 minutes to solve.

The $n=3$, $v=5$ case is solved in 193 minutes. Larger problems will be even more difficult to solve. However, the method can still be used for many practical problem sizes. Much faster processors than 700 MHz could be available, speeding up solutions somewhat. Typical problems will be in the $n=2$, $v=4$ size range. Wide Area Search Munitions are commonly dropped in pods of four, and four searching munitions will very rarely encounter three or more targets nearly simultaneously. For the rare occasion that three or more targets are encountered nearly simultaneously, the highest value targets could be assigned first in a 4 by 2 problem, and then the remaining tasks assigned to the remaining vehicles in a second assignment computation. Efficient suboptimal solutions can also be found for much larger problems by decomposing a larger problem into multiple smaller problems. A large problem with $n=4$, $w=8$ can be decomposed into two $n=2$, $w=4$ problems and solved in a few seconds. Finally, some simplifications to the problem structure can be made, that should greatly reduce problem complexity and solution times. For small T_{ii}, the vehicle that is assigned to classify a target is nearly always assigned to attack it as well. In this case, classify and attack can be grouped as a single task, substantially reducing the total number of tasks, variables, and constraints. Our future work will examine problem simplification and implementation into the MultiUAV simulation used in [1-3].

5   Conclusions

We have presented a method for using a Mixed Integer Linear Program (MILP) formulation to find the optimal solution to a multiple-task assignment problem where the tasks are coupled by timing and task order constraints. This formulation allows variation of vehicle flight paths to guarantee that timing constraints are satisfied, and directly incorporates the varying task completion times into the optimization. This is a promising formulation, which allows a true optimal solution for a very challenging problem. Solution results were presented for practical problem sizes, but scaling issues will require further work before the method can be applied to large problems. Future work will simply the problem structure to reduce complexity and apply the method to a task assignment problem in a detailed UAV simulation, including more realistic cost functions.

References
1.   Schumacher, C, Chandler, P. R, Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions Via Iterative Network Flow", Proceedings of the 2002 AIAA Guidance, Navigation, and Control Conference.
2.   Schumacher, C, Chandler, P. R, Rasmussen, S. J., "Task Allocation for Wide Area Search Munitions with Variable Path Length", Proceedings of the 2003 American Control Conference
3.   Schumacher, C, Chandler, P. R, Rasmussen, S. J., "Path Elongation for Task Assignment", Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference.
4.   Schumacher C., Pachter, M.,   Pachter, Chandler P., L.S., "UAV Task Assignment with Timing Constraints," Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference.
5.   Alighanbardi, M., Kuwata, Y., How, J., "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering," Proceedings of the 2003 American Control Conference.
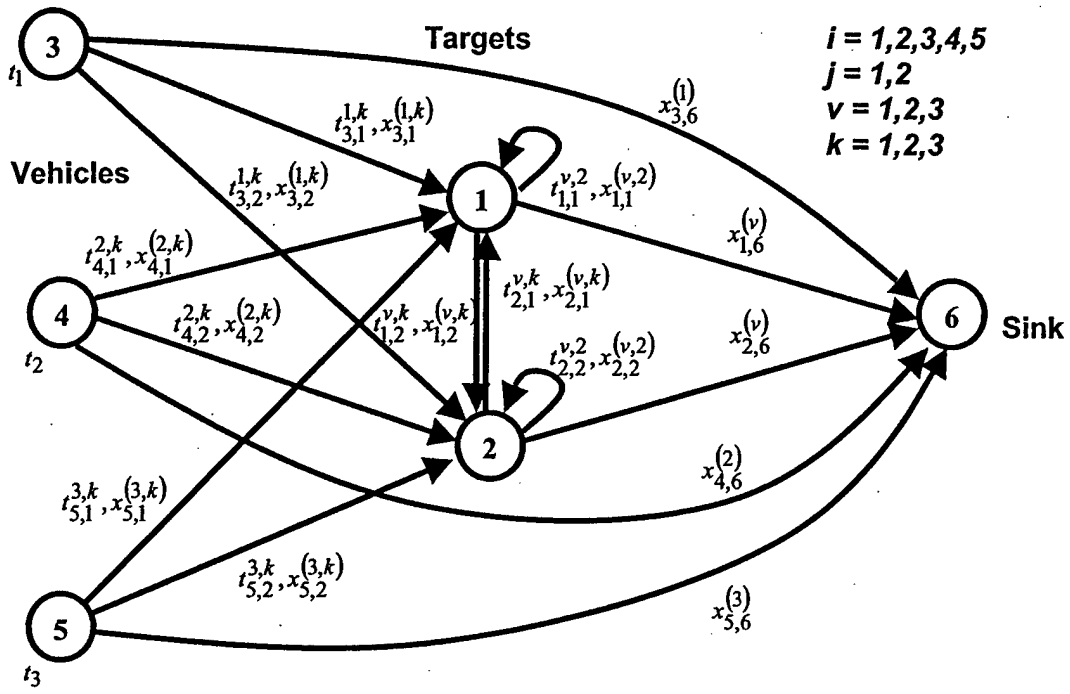6.   Nemhauser, G. L., and Wolsey, L.A., "Integer and Combinatorial Optimization", Wiley, 1988

Figure 1 – State Transition Diagram for 2 Targets, 3 Vehicles